A RAND NOTE

The Design of the MH Mail System

Robert H. Anderson, Norman Z. Shapiro,
Tora K. Bikson, Phyllis H. Kantar

December 1989

RAND

The research described in this report was conducted in RAND's Institute for Research on Interactive Systems.

Published by The RAND Corporation
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

A RAND NOTE

N-3017-IRIS

The Design of the MH Mail System

Robert H. Anderson, Norman Z. Shapiro,
Tora K. Bikson, Phyllis H. Kantar

December 1989

RAND

Institute for
Research on
Interactive Systems

PREFACE

Information scientists at RAND have had a continuing interest in the design and appropriate use of electronic mail systems. During the past decade, this interest has manifested itself in the design of the MH electronic mail system, widely distributed in many releases of the UNIX operating system. The original user's manual for MH was B. S. Borden, R. S. Gaines, and N. Z. Shapiro's *The MH Message Handling System: User's Manual,* The RAND Corporation, R-2367-AF, November 1979; guidelines for use of electronic mail systems were proposed in N. Z. Shapiro and R. H. Anderson's *Toward an Ethics and Etiquette for Electronic Mail,* The RAND Corporation, R-3283-NSF/RC, July 1985. Many MH users have exploited its power and adaptability without fully realizing the underlying source of that power. To date, the observations and principles underlying the design of MH have not been documented. This Note's purpose is to rectify that omission.

The authors think the design principles embodied in MH remain highly relevant and important for interactive information systems, yet many major systems—including recently developed electronic mail or "office information" systems—do not follow these principles (to their detriment, the authors believe). This Note should be of interest to designers, selectors, and users of interactive computer systems.

e-mail address (if the fragment is ambiguous, the ambiguity is displayed to the user for further clarification); and

- Automatic printing and routing, within corporate internal mail delivery, of messages to be delivered in hard-copy form.

Some e-mail systems tend to empower the user as sender; some empower the user as receiver. We believe that the MH system, although providing considerable flexibility and power to the user in both these roles, particularly empowers the user as a *processor* of information exchanged electronically within work groups. The design bias of MH can be summarized as "all power to the user," with both the costs and advantages that maxim entails.

In use in thousands of institutions worldwide, MH is distributed as part of many standard releases of the UNIX operating system. It is in the public domain.

CONTENTS

I. INTRODUCTION

Electronic mail (e-mail) will probably be an important component of any well-designed system for computer-supported cooperative work (CSCW). This proposition is evidenced by the many commercial and experimental systems for CSCW that have appeared over the years; in almost all cases, users have used them for electronic mail, sometimes by stretching and distorting the designers' intentions (Shapiro and Anderson, 1985).

This Note describes one e-mail system, MH, in use throughout The RAND Corporation for more than nine years. It is now in the public domain and used by thousands of other organizations. We present the principles and assumptions underlying MH's design, key architectural features that make MH effective for supporting cooperative work, and examples of features that have proved especially useful in our own corporate environment.

An electronic mail system: (1) permits the asynchronous electronic interchange of information between persons, groups of persons, and functional units of an organization; and (2) provides the mechanisms supporting the creation, distribution, consumption, processing, and storage of this information. Some—but not necessarily all—of this information will be structured. We emphasize its potentially unstructured aspect because we believe an essential attribute of e-mail (in addition to its asynchronicity) must be flexibility.

Finally, in our view, a highly desirable attribute of electronic mail—although not part of its essential nature—is heterogeneity. To be capable of evolutionary growth, systems should not require that identical, homogeneous computer hardware or software be used by all participants. This freedom is especially important for systems that span organizations or organizational boundaries. Heterogeneity is particularly critical for the long-term goal of integrating the work of many people, where each person uses his or her own favorite applications and is likely to be a member of multiple groups (Bikson, Gutek, and Mankin, 1987). In this environment, highly specific CSCW applications may not be desirable. We contend that examining which features of relatively generic systems make them suitable for supporting cooperative work among members of potentially diverse system environments is important. We examine the design of MH as an example of a good tool for collaboration in view of its flexibility, heterogeneity, and power for the user.

II. SOME OBSERVATIONS ABOUT ELECTRONIC MAIL

The architecture of MH was derived from basic observations about the nature of electronic mail. First, using electronic mail has much in common with other activities the user performs with a computer. The difference is that e-mail is a mechanism for dealing with the rest of the world, whereas most computer interfaces primarily involve communication between the user and programs within his or her own computer(s). This distinction is minor because it chiefly involves just wrapping an "envelope" around whatever other information activities the user has performed (see Talbert, Bikson, and Shapiro, 1984).

Second, a generic set of information manipulation operations exists, such as composition, storage, retrieval, and copying. Although we cannot specify here exactly what this set should be, we are certain that all the same operations also apply to electronic mail.

From these two general observations, we derive three design implications. First, the user-computer interface to information manipulation functions should be the same whether or not the user is working on e-mail. Adhering to this design principle creates an important benefit. If the same user interface tools are used for e-mail as for other information manipulation functions, then improvements to the interface (for example, providing graphic or windowing options) can automatically enhance the e-mail system as they become available. For example, as windowing environments such as SunView[1] have become available for UNIX,[2] we have routinely used those features as an "enhancement" to MH by using one window for an overview scan listing of message headers, another for message composition, and a third for alerting the user that new mail has been received. The move to a windowing environment provided considerably more power to the user in controlling simultaneous message system processes, but entailed *no changes at all* to the MH system itself. Other examples of our experience with this form of "automatic" improvement are described below.

Second, the *processes* used to perform information manipulation tasks should be the same whether or not the user is working on e-mail. For example, printer access, privacy control, priority assignment, accounting and the like should all use the same underlying computer processes.

---

[1]SunView is a registered trademark of Sun Microsystems, Inc.
[2]UNIX is a registered trademark of AT&T Bell Laboratories.

Third, a user's work life involves synthesis across various specific applications. Accessing a calendar, creating or retrieving bibliographic references, using a spreadsheet or word processor or Rolodex-type program, updating or using a corporatewide database, using a decision-support system, or sending a message containing fragments of information from these other activities—all these activities are not self-contained separate islands. Even if such activities are physically separate processes within a computer, they should appear to the user as a consistent set of information manipulation tools.

The main conclusion we have drawn for e-mail design is that it should not be an encapsulated, self-contained system providing its own interfaces and information-handling processes. Instead, to whatever extent possible, it should use existing resources for generic operations.

III. MH DESIGN AND RANDMAIL ENHANCEMENTS

MH DESIGN

The UNIX-based[1] e-mail system MH was developed in 1978 at RAND.[2] The design of MH tries to embody the implications outlined above through two main design decisions:

- MH commands—the primitive operations on a message—are UNIX shell commands; and
- Each MH message is a normal UNIX file.

From these decisions, it follows that collections of related messages may be placed into UNIX directories, which MH calls folders, as can folders of folders and so on because of UNIX's hierarchical file system. All normal UNIX file and directory operations are therefore available for use on MH messages. A file is the unit of information this operating system can handle. By making a message a file, then, we gain the power of the operating system on the essential unit of information in an e-mail system, a message. For space and operating efficiency, some e-mail systems use a file to store a *collection* of messages; MH sacrifices some of this efficiency for the advantages of the file = message equation.

Because of these design principles, users can, for example, specify (either in a profile of defaults, or at the time of message creation) a favorite text editor be used for message composition; the same editor used for creating other files is invoked for creating messages. Within the "e" editor commonly in use at RAND, any UNIX program or filter can be invoked with its (standard output) results inserted at the cursor's current location in a file. Thus, all the power of UNIX and its applications is directly available during the composition of a message and all user-supplied parts of its header. Users often concatenate a file into the

---

[1]The following description of MH design features uses UNIX terminology for consistency and because of MH's historical roots. However, these same design principles apply to the design of electronic mail systems within other operating systems having some of the modularity and flexibility of UNIX.

[2]The MH design was conceived by Norm Shapiro and Stockton Gaines at RAND circa 1977. The first version was implemented by Bruce Borden in three weeks under UNIX version 6 in late 1978, and was in RANDwide use within six months. In 1982, under the leadership of Marshall Rose at the University of California, Irvine, MH began a five-year series of metamorphoses. RANDmail enhancements were added at RAND in February 1984.

body of a message; or, within a reply, they may cut and paste portions of the message being answered. As more powerful word processors such as WordPerfect become available under UNIX, all their features similarly become available for message composition, editing, and so on under MH.

MH ENHANCEMENT

Use of the basic MH system became increasingly problematic as it expanded. As it escaped the confines of the initial user groups and spread more broadly, we learned that not all users

- Had easy access to terminals or personal computers; even those who did might prefer to receive e-mail messages in hard-copy form (either in addition to, or in place of, an electronic version);
- Knew the terminal access capabilities or media preferences of all other users;
- Resided on the same machine or file server;
- Knew all other users' log-in names or host machines so messages would be correctly addressed.

For these reasons, we enhanced MH with a system—RANDMail—tailored to RAND's organizational needs.[3] RANDMail is based on the theory that when you want to communicate with a person, the way you address that person should be independent of the communication's modality. That is, you should be able to look up someone's room number or telephone number, or give the name in the "To:" line of a message, in the same way. Furthermore, all reasonable descriptors of a person (for example, initials, nicknames, portions of a name) should be valid electronic mail addresses, just as they usually are in internal paper mail. To know log-in names, machine locations, or routes should not be necessary. In short, the system should be adaptable to the way groups work (Bikson, 1987).

[3]MH now runs under MS-DOS and a variety of UNIX versions and computer architectures; it is in use at several thousand sites. Organizations contributing heavily to MH include RAND; the University of California, Irvine; the University of California, Berkeley; and Northrop. Individual contributors included Diane Alexander, Robert Anderson, Cliff Bamford, Donna Betancourt, Tora Bikson, Bruce Borden, David Crocker,Terry Domae, Stockton Gaines, Van Jacobson, Phyllis Kantar, Mark LaCasse, John Romine, Marshall Rose, Norm Shapiro, Einar Stefferud, Jerry Sweet, Lee Talbert, and Terry West. The program is in the public domain.

To implement this premise, we made four additions to MH. First, we created a single corporatewide database, which all RAND computers could access, containing for each company employee or consultant information fields such as name (plus nickname, if any), extension, department, mail stop, log-in name, home computer, and message routing (that is, whether the message was to be sent hard copy or soft copy, or both).

Second, we created a new UNIX command, *name*, to access this database. Followed by any unambiguous abbreviation of a person's full name, the *name* command generates a listing of the entire database entry for that person. A *name* command followed by fragmentary information (that is, ambiguous within the database) results in an overview "scan" listing showing summary information for all individuals in the database meeting the criteria. For example, *name RA* results in the display of several names (Ruth Almond, Robert H. Anderson, Rae Archibald) satisfying that pattern.

Third, anywhere a recipient's name can appear within a message header (for example, in the "To:," "cc:," or "bcc:" fields), giving an identifier that uniquely identifies the individual within the database in the same form as an argument to the *name* command is sufficient. A message header, for instance, might be composed as "To: RHA, PKantar, NShap."

The system would expand this header into complete proper names (with computer addresses) by referring to the database at the time the message was sent. If any abbreviated name is ambiguous, the user receives a listing such as the one described above, with the option to re-edit the message to correct the ambiguity.

Fourth, everyone in the organization can receive an electronic message, regardless of terminal/workstation access. Messages addressed to users who need or prefer hard copies are automatically routed to a printer and are delivered in the next internal mail distribution.

By itself, each of these features is straightforward. Together, they mean something very important: Every message is routed to a *person*. As a person changes rooms, departments, host computers, name (for example, from maiden to married) and the like, a single update to a master database assures that the person gets the message. Significantly, the same corporate database is used to print the corporate telephone directory; it is also used by RAND telephone operators to route calls to primary or auxiliary telephone extensions. As a central corporate data facility, its chances for accuracy and timeliness are greatly increased. And although we call the enhanced system RANDMail, we think its generic features are not unique to RAND. Any organization striving for computer-supported cooperative work would do well to create an on-line database about people and their preferences to facilitate the exchange of electronically captured information.

IV. RANDMAIL FOR COMPUTER-SUPPORTED COOPERATIVE WORK

We have suggested that combining the power of UNIX with RANDMail primitives as described above will support a variety of needs related to computer-supported cooperative work. Below, we show how MH facilities compare with other approaches to satisfying those needs. For convenience, we have grouped the facilities as (1) the user as sender, (2) the user as receiver, and (3) the user as mail processor.

THE USER AS SENDER

A main strength of The Coordinator[1] (Winograd and Flores, 1986; Flores, 1982) is the power it gives the message's sender. He or she can determine the type of message (for example, whether it is part of a conversation for action or for possibilities), which in turn determines the type of follow-up processing performed in both the sender's and the recipient's systems.

In attempting to provide similar power to the "user as sender," MH would rely on users' capabilities to tailor their messaging environment (for example, in the mail profile). For instance, a user may want to tailor the message system to facilitate a common group operation, such as establishing a milestone task to be completed by a certain date; one method in MH is to add certain fields to a standard message form, such as "Msg-type: task" and "Completion-date:." The user's *.login* file could then contain a standard command to be issued upon each log-in, such as *check-completions*. This command file could test all messages of type "task" in the user's current folder, for instance, and give notice of completion dates earlier than the day's date for which no corresponding reply had been received.

THE USER AS RECEIVER

In contrast, the information lens model (Malone et al., April 1987 and May 1987) seems to emphasize facilities for the user as message receiver. In MH, similar kinds of power to the user as receiver would also be provided through shell files. Perhaps the best example of this is what we have come to call "message triaging." A UNIX shell file of commands is created that performs MH *pick* operations to identify incoming messages, for

---

[1]The Coordinator is a registered trademark of Action Technologies, Inc.

example, as being from certain correspondents, or having certain keywords in their subject lines, or coming from a certain institution. These messages are automatically identified using standard features of *pick;* the messages may then be refiled from the MH in-box to other mail folders. As time permits, the user may then look through folders labeled "urgent," "routine," or perhaps having the names of specific projects on which the user works. Such processing of received messages may be quite independent of any knowledge or cooperation by the sender(s). For instance, if "Dave L." is your boss, you may send all messages from him automatically to the "urgent" folder. Of course, the triaging of incoming messages is facilitated if senders within a work group observe some standard protocols, such as using specific project words in the subject line to indicate message content. But to assure power over the handling of incoming messages independent of sender involvement, MH avoids rigid subject-line requirements.

THE USER AS MAIL PROCESSOR

If MH has an emphasis, it is probably on providing facilities to users as general processors of mail. This orientation accords well with work structures at RAND, where individuals typically belong to multiple work groups; groups form and reform relatively quickly; and individuals are quite likely to be a leader of one work group and a subordinate in another. A sampler of MH practices illustrating this orientation in the RAND environment follows.

Suppose, for instance, your group wants to code certain messages as belonging to a category (for example, related to the keyword *proposal*), so they can be stored, located, or referenced together. The easy way to do this is to agree within the group always to use a keyword or phrase such as *proposal* within the subject line. Then the MH *pick* command can be used to access all such messages in your electronic in-box as follows:

```
% pick -subject proposal
```

To pick all such proposal-relevant messages since last Friday and refile them into a folder called "prop" while obtaining a scan listing of the messages selected, you could issue these two commands:

```
% refile +prop `pick -subject proposal -after friday`
% scan  +prop
```

The *pick* command extracts the requested messages and returns a "message sequence," or list of the message numbers satisfying the request. That message sequence becomes an input parameter to the *refile* command, which files them in the folder (a normal

UNIX subdirectory) prop. A scan command for the prop folder then produces a "scan listing" with one line per message, giving an overview of the folder's contents. The connectives *-and, -or, -not* along with braces can be used with the *pick* command to create Boolean conditions. In addition, a regular expression of the form used by UNIX's *grep* command can be used to indicate the string to be searched for within a field—or anywhere within the message.

Rather than having everyone within a work group remember to use a keyword within the subject line of a message, creating prepared forms for messages is often easier; these forms can have additional fields built into their headers, which *pick* can access. For example, in composing a message, the user can issue a "form" flag telling which UNIX text file to use as the beginning "message form":

% comp -form propgroup

The UNIX *propgroup* file might have a prebuilt message header containing additional fields, such as "Keyword: proposal." Anyone using this form could then pick all proposal-related messages from the current folder by issuing the *pick* command:

% pick - -Keyword proposal

The double dash here indicates a search for a nonstandard field name within the header, followed by any regular expression indicating a search string in that field. The prepared message form might also have a prebuilt "cc:" line, if a standard routing list for these messages exists.

Note that the group-customized message header, plus unlimited room for the message body, is just a standard UNIX text file that comes up within a text editor window. The header's contents may be revised *at any time* during the composition of a message, a "surprisingly useful" feature (apologies to Malone et al., April 1987) because changing one's mind about the distribution list, subject line, and so on as a message takes shape is easy. Further, UNIX command (shell) file features can be used to abbreviate frequently used combinations of message commands, so that commonly used sequences can be invoked by a simple identifier.

Many other examples of flexibility empowering the generic mail user in MH could be given. Those provided here were chosen in an attempt to give concrete illustrations of the design principles that engendered them. A quantitative study of the spread of the RANDMail system throughout RAND, showing patterns of communication and many other aspects of its use, is contained in Bikson (1987) and Eveland and Bikson (1987). A description of users' experience with MH and other electronic mail systems and resulting user guidelines is available in Shapiro and Anderson (1985).

V. CONCLUSIONS

To date, the design and use of MH and its RANDMail extensions have been guided by several principles:

- What people do when handling electronic mail is mainly what they do anyway: create files, edit text, group related information in directories, search for information, and delete files. An electronic mail system should build upon existing tools for these tasks—tools that are known and comfortable to the users.
- Messages are for people. The same names, nicknames, and common references used in other media should be valid in addressing electronic messages.
- "All power to the user," whether as sender, receiver, or processor, is a good design maxim. Rather than providing a fixed message header, or fixed types of message forms, a system should allow users to create the message header fields they need and then perform the desired processes on these fields and their contents.
- The totality of a message, including any user-specifiable portions of its header, should be changeable by the user at any time during message creation or editing. The development of the header and the message body are interrelated acts and should be treated as one conceptual unit.
- Achieving mail system functionality at the expense of flexibility and heterogeneity is not necessary. A generic mail system can be accommodated and tuned to support a specific work-group environment, as we have demonstrated by wedding the features of UNIX and MH primitives.

Electronic mail systems, as everything else, have trade-offs. With MH or other systems designed according to the principles we have suggested, generic functionality and certain simple defaults are immediately available to all users. To attain more substantial advantages and to take full advantage of providing "all power to the user," users must be able to invest time and effort in learning about the system and in developing their communication environments. The bad news is that when the effort is not made, work

groups do not end up with a system customized to fit their work environments; the good news is the adaptability. If your project team changes tomorrow, you can change your mail environment accordingly; if you want to change the "cc:" line the minute you change your mind about who should be copied, you can—and so on. MH implements the "all power to the user" philosophy we find in UNIX, with the work-group costs and advantages that maxim entails.

BIBLIOGRAPHY

Bikson, T. K., *New Office Technology: Planning for People,* Work in America Institute, Studies in Productivity Series, Pergamon Press, New York, 1986.

Bikson, T. K., "Understanding the Implementation of Office Technology," in *Technology and the Transformation of White-Collar Work*, edited by R. Krant, Lawrence Erlbaum Associates, Hillsdale, N.J., 1987; also, The RAND Corporation, N-2619-NSF, 1987.

Bikson, T. K., B. A. Gutek, and D. A. Mankin, *Implementing Computerized Procedures in Office Settings: Influences and Outcomes*, The RAND Corporation, R-3077-NSF/IRIS, 1987.

Borden, Bruce S., R. Stockton Gaines, and Norman Z. Shapiro, *The MH Message Handling System: User's Manual,* The RAND Corporation, R-2367-AF, November 1979.

Eveland, J. D., and Tora K. Bikson, "Evolving Electronic Communication Networks: An Empirical Assessment," *Office: Technology and People,* Vol. 3, 1987, pp. 103–128.

Flores, Fernando, *Management and Communication in the Office of the Future,* Logonet, Berkeley, Calif., 1982.

Malone, Thomas W., et al., "Semi-Structured Messages Are Surprisingly Useful for Computer-Supported Coordination," *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, April 1987, pp. 115–131.

Malone, Thomas W., et al., "Intelligent Information-Sharing Systems," *Communications of the ACM*, Vol. 30, No. 5, May 1987, pp. 390–402.

Rose, M. T., E. A. Stefferud, and J. N. Sweet, "MH: A Multifarious User Agent," *Computer Networks,* Vol. 10, No. 2, September 1985, pp. 65–80.

Rose, M. T., and John L. Romine, "MH.5: How to Process 200 Messages a Day and Still Get Some Real Work Done," Proceedings, Summer Usenix Conference and Exhibition, Portland, Oregon, June 1985, pp. 455–487.

Shapiro, Norman, and Robert H. Anderson, *Toward an Ethics and Etiquette for Electronic Mail,* The RAND Corporation, R-3283-NSF/RC, July 1985.

Talbert, L., T. K. Bikson, and N. Z. Shapiro, *Interactive Information Environments: A Plan for Enabling Interdisciplinary Research*, The RAND Corporation, N-2115, 1984.

Winograd, Terry, and Fernando Flores, *Understanding Computers and Cognition: A New Foundation for Design,* Ablex, Norwood, N.J., 1986.